The Ultimate Solar Azimuth Formula: A note on the formula that renders circumstantial treatment unnecessary and an update of the ephemerides to that of *The Astronomical Almanac for the Year 2019*

Taiping Zhang[a,*], Paul W. Stackhouse Jr.[b], Bradley Macpherson[c], J. Colleen Mikovitz[a]
[a]SSAI/NASA Langley Research Center, One Enterprise Parkway, Suite 200, VA 23666-5845, USA
[b]NASA Langley Research Center, Mail Stop 420, Hampton, VA 23681-2199, USA
[c]GSFC/Booz Allen Hamilton, Norfolk, VA

Abstract

A conceptually and mathematically concise formula for computing the solar azimuth angle has been used by a subgroup of scientists, but for lack of documentation and publication, it has not been well circulated. This note introduces this formula which is based on the idea of a unit vector, $\boldsymbol{S}$, originating from the observer's location and pointing toward the center of the Sun. The vector is completely determined by the coordinates of the *subsolar point* and of the observer. The x- and y-components of the vector determine the solar azimuth angle, and their use along with the function **atan2**, which is available in a number of programming/scripting languages, including **Fortran** and **Python**, renders any circumstantial treatment absolutely unnecessary. The z-component of the vector, at the same time, determines the solar zenith angle.

1 Introduction

We recently modified a code that computes the solar irradiance on tilted surfaces. In this process, we reviewed available formulas for computing the solar azimuth angle in particular. Typically, there are three formulas expressing the sine, cosine and tangent of the solar azimuth angle as functions of the declination and the zenith angle of the Sun, and the latitude and hour angle of the observer's point, and they can be found in nearly all the listed references. Each of these formulas needs circumstantial treatment, which can be ponderous and not easy to understand, to put the result from the inversion of a trigonometric function into the right quadrant. Probst (2002), Sproul (2007), Bhatia (2014) and Soulayman (2018) provide warnings about using these formulas and discussed at length how the treatment is performed.

Braun and Mitchell (1983) used the sine of the solar azimuth angle and devised a sophisticated algorithm to put the result in the right quadrant. In fact, our original code was based on this algorithm. Test runs, however, show that the algorithm gives correct results only in the Northern extratropics.

* Corresponding author: Tel: +1 757-951-1904, E-mail address: Taiping.Zhang@NASA.gov (T.Zhang).

Neither the sine nor the cosine function alone can uniquely determine the solar azimuth angle, since neither is an injective, or one-to-one, function in the range [-π, π], or [-180°, 180°]. The sine and cosine functions joined together as a pair, on the other hand, can uniquely determine the solar azimuth angle. However, dividing the sine function by the cosine function, which creates the tangent function, causes the information to degenerate, as the tangent function is not a one-to-one function in the full possible range of the azimuth angle, which is why the value from the arctangent function also needs further treatment to get the final angle in the right quadrant. What is provided in *The Nautical Almanac* (2019) is one using the tangent function.

Meanwhile, a subgroup of scientists has been using a formula based on the x-, y- and z-components of a unit vector, ***S***, which is entirely determined by the coordinates of the observer and the *subsolar point*, and the unit vector originates from the observer's coordinates, or the latitude and longitude, and points toward the Sun. Now the solar zenith angle is just the **arccosine** of the z-component, and the solar azimuth angle is simply the **atan2** of the x- and y-components, and the **atan2** here is a function that takes two separate arguments and is available in such as **Fortran** and **Python**. Aside from the simplicity of this method, the *equation of time* in conveniently incorporated into the expression for the longitude of the subsolar point. It is not clear, however, who first derived the formula and, in addition, for lack of documentation and publication, it has not been widely circulated. Coincidentally, Sproul (2007) derived the said unit vector using vector analysis and obtained identically the same x-, y- and z-components mentioned above; from there, however, the author did not proceed to use the method discussed here.

The variables involved in deriving solar geometry can be put into two categories: 1) those that fall within the area of astronomy and can be termed, appropriately, *ephemerides*, and the relevant ones in this category are the declination of the Sun, the Earth-Sun distance, and the equation of time; 2) those that purely belong to 3D geometry and/or spherical trigonometry, and the ones in this category are solar zenith and azimuth angles and can be derived from those in Category 1).

For first approximations, there is a simple formula for the declination of the Sun which neglects the year-to-year change; the Earth-Sun distance is considered a constant, namely, 1 au; and the *equation of time* is ignored. Since it is easy and straightforward to use those formulas in *The Astronomical Almanac*, as has been done by Michalsky (1988), there is not a good reason to opt for simpler ones with more uncertainties. In this note, we will first provide formulas from *The Astronomical Almanac for the Year 2019* for variables in Category 1), and then give formulas for those in Category 2). To illustrate how these formulas work, we put some results in two figures. Finally, our subroutine in **Fortran 90** is put in Appendix A.

2 The Formulas

2.1 The declination of the Sun (δ), the Earth-Sun distance (R), and the equation of time (E)

The following formulas are straight from *The Astronomical Almanac for the Year 2019* except the time argument n has been modified for an arbitrary year, preferably from 1950 to 2050, and they are in the order such that results from earlier expressions can be used by later expressions. Only those equations of immediate interest are numbered.

$n = -1.5 + (Y_{in} - 2000) \cdot 365 + N_{leap}$+ Day of Year + Fraction of Day from 0:00 UT (day),
$L = 280.466 + 0.9856474n$ (°),
$g = 357.528 + 0.9856003n$ (°),
$\lambda = L + 1.915 \sin g + 0.020 \sin(2g)$ (°),
$\varepsilon = 23.440 - 0.0000004n$ (°);
$\alpha = \tan^{-1}(\cos \varepsilon \tan \lambda) \cdot 180/\pi$ (°),
$\delta = \sin^{-1}(\sin \varepsilon \sin \lambda) \cdot 180/\pi$ (°), (1)
$R = 1.00014 - 0.01671 \cos g - 0.00014 \cos(2g)$ (au), (2)
$E_{min} = (L - \alpha) \cdot 4$ (min). (3)

where
n is the number of days of Terrestrial Time (TT) from J2000.0 UT;
Y_{in} is the input year;
N_{leap} is the number of leap years;
L is the mean longitude of the Sun corrected for aberration;
g is the mean anomaly;
λ is the ecliptic longitude;
ε is the obliquity of ecliptic;
α is the right ascension;
δ is the declination of the Sun;
R is the Earth-Sun distance;
E_{min} is the equation of time.

Note that L and g as well as λ given as above can be either positive or negative, but computationally they need to be put in the range 0° to 360°, and this can be accomplished by using the **modulo** function; α needs to be in the same quadrant as λ, and this can be done by using the **atan2** function, which takes two arguments, instead of the **atan** function, which takes only one argument. All these treatments have been properly taken care of in the code in Appendix A.

According to the *Almanac*, the errors of the right ascension and declination of the Sun given by these formulas are less than (1/60)°, and the error of the equation of time is less than 3.5 seconds, if the input year is between 1950 and 2050.

2.2 The solar zenith angle and azimuth angle

Suppose the observer's coordinates, or latitude and longitude, are (ϕ_o, λ_o), and the *subsolar point*'s coordinates are (ϕ_s, λ_s), then the x-, y- and z-components of the unit vector, $\boldsymbol{S}$, pointing from the observer to the center of the Sun are as follows:

$$\phi_s = \delta, \tag{4}$$
$$\lambda_s = -15(T_{GMT} - 12 + E_{min}/60), \tag{5}$$
$$S_x = \cos\phi_s \sin(\lambda_s - \lambda_o), \tag{6}$$
$$S_y = \cos\phi_o \sin\phi_s - \sin\phi_o \cos\phi_s \cos(\lambda_s - \lambda_o), \tag{7}$$
$$S_z = \sin\phi_o \sin\phi_s + \cos\phi_o \cos\phi_s \cos(\lambda_s - \lambda_o). \tag{8}$$

It can be shown that there exists $S_x^2 + S_y^2 + S_z^2 = 1$. Sproul (2007) used vector analysis to derive the x-, y- and z-components of $\boldsymbol{S}$, and they are exactly the same as the S_x, S_y and S_z here, noticing that $\lambda_s - \lambda_o$ differs from the hour angle, ω, by only a negative sign.

The solar zenith angle is now simply

$$Z = \cos^{-1} S_z, \tag{9}$$

and the solar azimuth angle following the *South-Clockwise convention* is

$$\gamma_s = \textbf{atan2}(-S_x, -S_y). \tag{10}$$

Eq. (10) gives an unambiguous solar azimuth angle and it is final, and it works everywhere from pole to pole. In other words, the solar azimuth angle is in the right quadrant. The standard **atan2**(y, x) function which follows the *East-Clockwise convention* is available in programming/scripting languages **Fortran**, **Python**, etc. and it gives the angle in the range [-π, π] which can be converted to [-180°, 180°].

The *North-Clockwise convention* can be achieved using $\gamma_s =$ **atan2**(S_x, S_y), and the *East-Counterclockwise convention* can be realized using $\gamma_s =$**atan2**(S_y, S_x), though the latter is rarely used in solar energy analysis.

The advantage of Eq. (10) is that it is not only concise, but never fails. When both S_x and S_y are 0, namely, when the subsolar point and the observer's point are the same and, therefore, the azimuth angle is not defined, Eq. (10) gives 0 and keeps the job running. This treatment is physically inconsequential, because the term involving azimuth angles in solar energy analysis also becomes 0.

The *subsolar point* is not an unfamiliar concept, but the way it is used as in the method discussed here dispels some of the mysterious aura around solar geometry, and the Earth-Sun configuration is thus mentally more accessible.

It can be easily shown that $S_y/\sqrt{S_x^2 + S_y^2}$, $S_x/\sqrt{S_x^2 + S_y^2}$ and S_y/S_x are equivalent to the three staple formulas traditionally used, namely, the sine, cosine and tangent of the azimuth angle, respectively. Both the signs of S_x and S_y are required to determine the azimuth angle, but each of the three formulas causes two possible distinct cases degenerate into one, which is why none alone can give an unambiguous result. Sproul (2007) derived the expressions of S_x, S_y and S_z using vector analysis, but did not use Eq. (10) to compute the solar azimuth angle; rather, the author proceeded to discuss the three traditionally formulas at length.

Note these formulas are based on geocentric latitudes and no correction for the atmospheric refraction has been applied to the solar zenith angle, Z. The atmospheric refraction depends on the temperature profile of the atmosphere and surface elevation and can be evaluated later.

Note that the equation of time, E_{min}, appears in the expression for λ_s, the longitude of the *subsolar point*.

A word of warning is that in computer languages, the input angles of trigonometric functions and output angles of inverse trigonometric functions are all in radians.

Finally, Eq. (10) is not limited to the computer. The idea behind it is applicable even when one uses a calculator, and it can be done as follows: 1) Calculate the values of S_x and S_y; 2) Determine which quadrant the angle falls in by inspecting the signs of S_x and S_y; 3) If the angle is in the 1st and 4th quadrants, $\gamma_s = \tan^{-1}(S_y/S_x)$; otherwise, $\gamma_s = \tan^{-1}(S_y/S_x) + 180°$, supposing the calculator is in the **degree** mode.

2.3 When the observed body is a satellite

In last section, it has been presumed that the Sun is a body with an altitude of infinity. In the case of a satellite with a finite altitude, H, it can be shown that Eqs. (6)-(10) can still be used, assuming now that (ϕ_s, λ_s) are the coordinates of the *sub-satellite point* and known, except an ΔZ_H needs to be added to the Z from Eq. (9) and, ΔZ_H is as follows

$$\Delta Z_H = \tan^{-1}\left[\frac{R \sin Z}{R(1-\cos Z)+H}\right], \tag{11}$$

where R is the radius of the Earth, assuming a spherical Earth, and H is the altitude of the satellite. The zenith angle of the satellite is thus $Z + \Delta Z_H$. Note that when H is infinity, $\Delta Z_H = 0$.

For completeness, the x-, y- and z-components of the unit vector originating from the observer and pointing toward the satellite are

$$S_x^{sat} = S_x \frac{\sin(Z+\Delta Z_H)}{\sin Z}, \tag{12}$$

$$S_y^{sat} = S_y \frac{\sin(Z+\Delta Z_H)}{\sin Z}, \tag{13}$$

$$S_z^{sat} = S_z \frac{\cos(Z+\Delta Z_H)}{\cos Z}, \tag{14}$$

where S_x, S_y, S_z and Z are from Eqs. (6)-(9), respectively, assuming that that (ϕ_s, λ_s) are the coordinates of the *sub-satellite point* and known. And the azimuth angle of the satellite is

$\gamma_{sat} = \mathbf{atan2}(-S_x^{sat}, -S_y^{sat}).$ (15)

Since S_x^{sat} and S_y^{sat} in Eq. (15) differ from S_x and S_y in Eq. (10) by the same factor, they give the same result.

2.4 When the observer is a satellite and the observed body is the Sun

Now assume that the coordinates of the *sub-satellite point* and the *subsolar point* are (ϕ_o, λ_o) and (ϕ_s, λ_s), respectively, Eqs. (6)-(10) can be used for the solar zenith angle and solar azimuth angle from the perspective of the satellite, assuming the satellite uses the terrestrial convention for directions. The exception is that the solar zenith angle at sunset is now $\frac{\pi}{2} + \cos^{-1}(\frac{R}{R+H})$, where R is the radius of the Earth, assuming a spherical Earth, and H is the altitude of the satellite. In the case of a surface-based observer, $H = 0$, and the solar zenith angle at sunset is simply $\frac{\pi}{2}$ in the absence of the atmospheric refraction.

3 Sample Results

Shown in Fig. 1a and 1b are the declination of the Sun, δ, the solar constant factor, $1/R^2$, and the equation of time, E_{min} for the year 2020 at 1-day step. These are directly from the *The Astronomical Almanac for the Year 2019*.

Fig. 1c shows the annual path of the subsolar point at GMT hours 8:00 to 13:00 at 1-hour interval in the year 2020. The “8” figures are due to the equation of time. Fig. 1d shows the x- and y-components of the unit vector, ***S***, at a site in the US. The rightmost “8” figure gets distorted the most, because the corresponding hour is the farthest from the local noon, and because these “8” figures are actually the horizontal projections of the analemmas shown in Fig. 2.

Fig. 2 shows the x-, y- and z-components of the unit vector, ***S***, throughout the year 2020 at 1-hour step at the same site as in Fig. 1c and 1d. This is a complete depiction of the position of the Sun for an entire year, because the z-component determines the solar zenith angle and the x- and y-components determine the solar azimuth angle. As the vector, ***S***, moves in time, its tip traces out the familiar “8” figures, and these figures are literally the analemmas that can be observed in the celestial sphere. This type of figure can be generated for any given location, so that the Sun’s position gets visualized in practically the whole domain of time.

4 Summary

This note presented a solar azimuth angle formula derived from a somewhat different perspective. The result is an algorithm that is conceptually more concise and computationally much simpler than the three traditional formulas, yet the algorithm has the same mathematical rigor and accuracy as any of the three traditional formulas. This algorithm also leads to better visualization of the annual excursion of the Sun's position for any given terrestrial location.

Acknowledgments

This work is part of the Prediction Of Worldwide Energy Resources (POWER) Project funded through the NASA Earth Science/Applied Science Program.

References

The Astronomical Almanac for the Year 2019. The United States Naval Observatory.

Bhatia, S.C., 2014. Advanced Renewable Energy Systems, Part I. Woodhead Publishing India Pvt. Ltd.

Michalsky, J.J., 1988. The Astronomical Almanac's Algorithm for Approximate Solar Position (1950-2050). *Sol Energy*, **40**(3), 227-235.

The Nautical Almanac, 2019. HM Nautical Almanac Office.

Probst, O., 2002. The apparent motion of the Sun revisited. *Eur. J. Phys.* **23**, 315–322.

Soulayman, S., 2018. Comments on solar azimuth angle. *Renew Energ*, **123**, August 2018, 294-300.

Sproul, A. B., 2007. Derivation of the solar geometric relationships using vector analysis. *Renewable Energy*, **32**, 1187-1205.

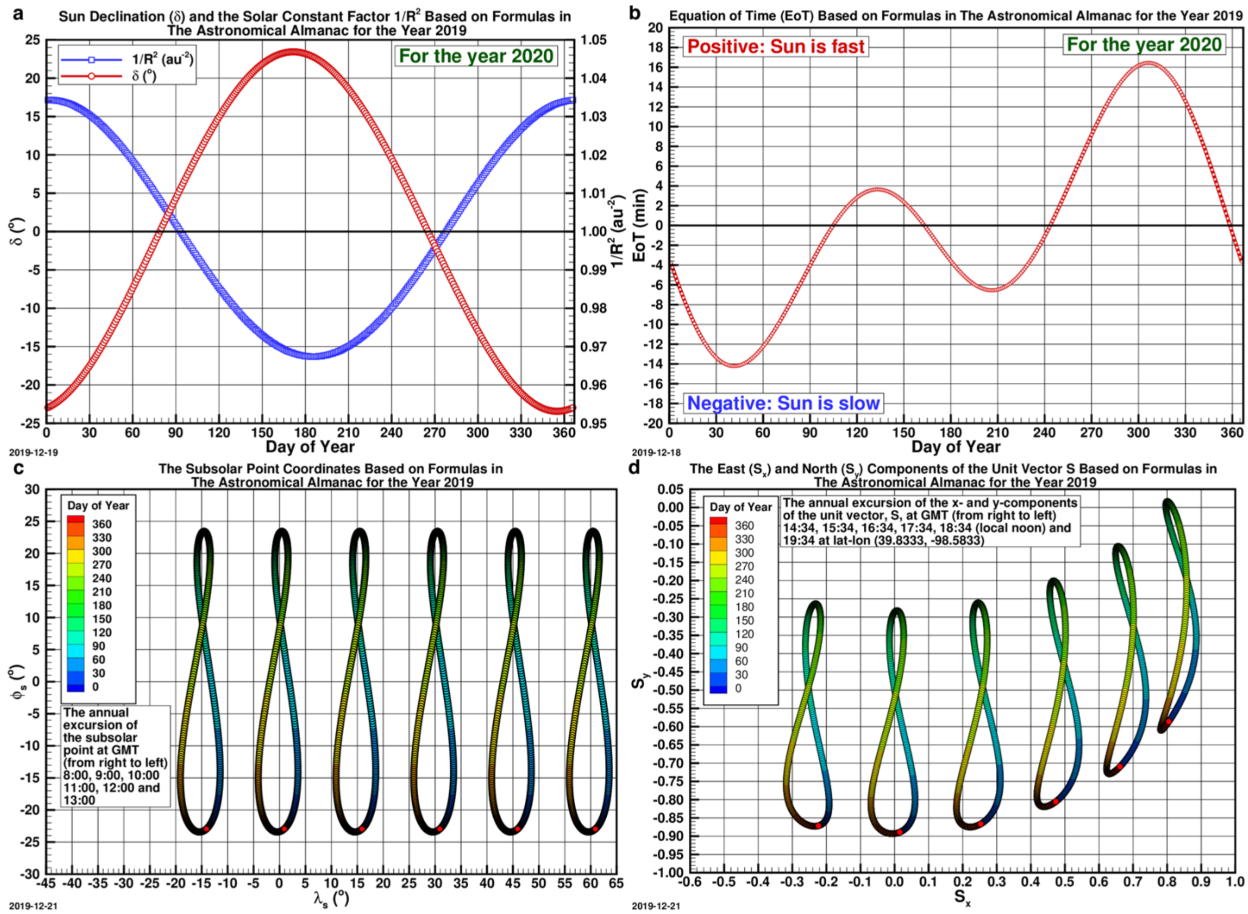

Fig. 1. a. The declination of the Sun, δ, and the solar constant factor, $1/R^2$ for the year 2020 at 1-day step; b. The equation of time, E_{min}, for the year 2020 in 1-day step; c. The track of the subsolar point at GMT 8:00, 9:00, 10:00, 11:00, 12:00 and 13:00 throughout the year 2020; d. The x- and y-components of the unit vector, S, at a site in the US site at GMT 14:34, 15:34, 16:34, 17:34, 18:34 (around local noon) and 19:34.

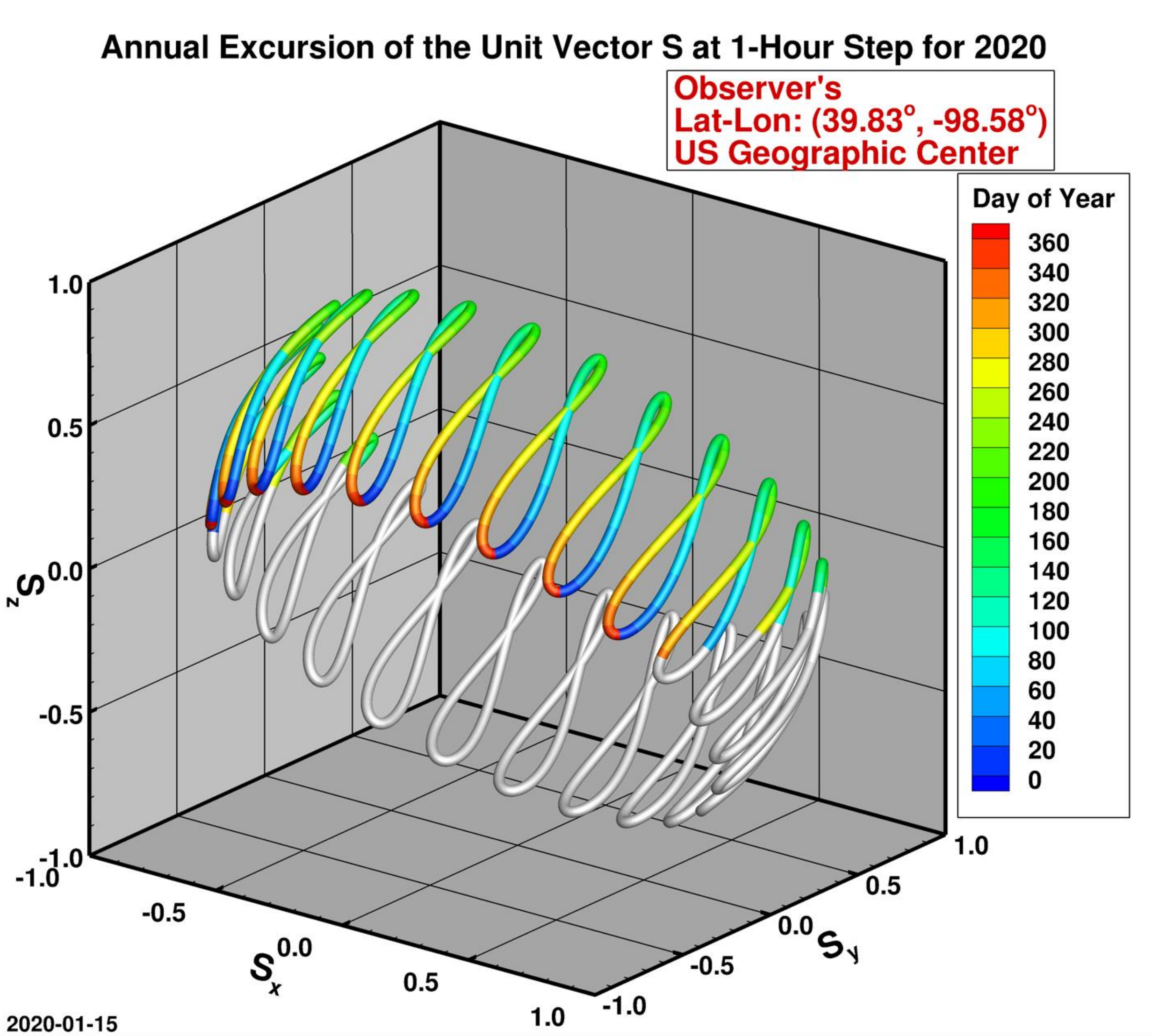

Fig. 2. Annual excursion of the unit vector ***S*** at 1-hour step throughout the year 2020 at a US site. Note that nighttime is in gray color and occurs when $S_z < 0$. Each figure "8" corresponds to the same hour and is literally an analemma in the celestial sphere. The "8" figure at the lowest position represents the hour around local midnight, and the hour progresses counterclockwise. Note that the color bar also applies to nighttime analemmas.

Appendix A: Subroutine in Fortran 90

```
! 2019-12-26  !Solar Geometry using subsolar point and atan2.
!             by Taiping Zhang.
! Input variables:
!    inyear: 4-digit year, e.g., 1998, 2020;
!    inmon: month, in the range of 1 - 12;
!    inday: day, in the range 1 - 28/29/30/31;
!    gmtime: GMT in decimal hour, e.g., 15.2167;
!    xlat: latitude in decimal degree, positive in Northern Hemisphere;
!    xlon: longitude in decimal degree, positive for East longitude.
!
! Output variables:
```

```
!    solarz: solar zenith angle in deg;
!    azi: solar azimuth in deg the range -180 to 180, South-Clockwise
!         Convention.
!
! Note: The user may modify the code to output other variables.

Subroutine sunpos_ultimate_azi_atan2(inyear, inmon, inday, gmtime, &
    xlat, xlon, solarz, azi)

  implicit none

  integer:: inyear, inmon, inday, nday(12), julday(0:12), xleap, i, &
      dyear, dayofyr
  real:: gmtime, xlat, xlon
  real:: n, L, g, lambda, epsilon, alpha, delta, R, EoT
  real:: solarz, azi, toadwn
  real, parameter:: rpd=acos(-1.0)/180
  real:: sunlat, sunlon, PHIo, PHIs, LAMo, LAMs, Sx, Sy, Sz
  data nday / 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 /

  if((mod(inyear, 100)/=0 .and. mod(inyear, 4)==0) .or. &
      (mod(inyear, 100)==0.and.mod(inyear, 400)==0)) then
    nday(2)=29
  else
    nday(2)=28
  endif

  julday(0)=0
  do i=1, 12
    julday(i)=julday(i-1)+nday(i)
  enddo
! Note: julday(12) is equal to either 365 or 366.

  dyear=inyear-2000
  dayofyr=julday(inmon-1)+inday
  if(dyear<=0) then
    xleap=int(real(dyear)/4)   !xleap has the SAME SIGN as dyear. !!!
  else
    xleap=int(real(dyear)/4)+1 !Since 2000 is a leap year.
  endif

! --- Astronomical Almanac for the Year 2019, Page C5 ---
  n=-1.5+dyear*365.0+xleap*1.0+dayofyr+gmtime/24
  L=modulo(280.460+0.9856474*n, 360.0)
  g=modulo(357.528+0.9856003*n, 360.0)
  lambda=modulo(L+1.915*sin(g*rpd)+0.020*sin(2*g*rpd), 360.0)
  epsilon=23.439-0.0000004*n
    alpha=modulo(atan2(cos(epsilon*rpd)*sin(lambda*rpd), &
        cos(lambda*rpd))/rpd, 360.0) !alpha in the same quadrant as lambda.
  delta=asin(sin(epsilon*rpd)*sin(lambda*rpd))/rpd
  R=1.00014-0.01671*cos(g*rpd)-0.00014*cos(2*g*rpd)
  EoT=modulo((L-alpha)+180.0, 360.0)-180.0 !In deg.

! --- Solar geometry ---
  sunlat=delta  !In deg.
  sunlon=-15.0*(gmtime-12.0+EoT*4/60)
```

```
  PHIo=xlat*rpd
  PHIs=sunlat*rpd
  LAMo=xlon*rpd
  LAMs=sunlon*rpd
  Sx=cos(PHIs)*sin(LAMs-LAMo)
  Sy=cos(PHIo)*sin(PHIs)-sin(PHIo)*cos(PHIs)*cos(LAMs-LAMo)
  Sz=sin(PHIo)*sin(PHIs)+cos(PHIo)*cos(PHIs)*cos(LAMs-LAMo)

  solarz=acos(Sz)/rpd      !In deg.
  azi=atan2(-Sx, -Sy)/rpd  !In deg. South-Clockwise Convention.

Endsubroutine sunpos_ultimate_azi_atan2
```